jian-jia.chen [☺] tu-dortmund.de
ingo.korb [☺] tu-dortmund.de

Exercises for Lecture
Embedded Systems
Winter Semster 16/17

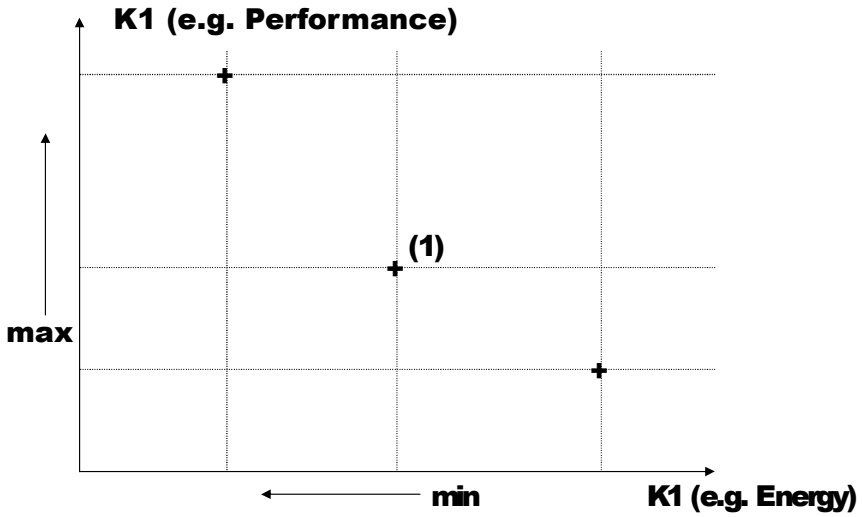
Written Exercise Sheet 6

(15 Punkte)

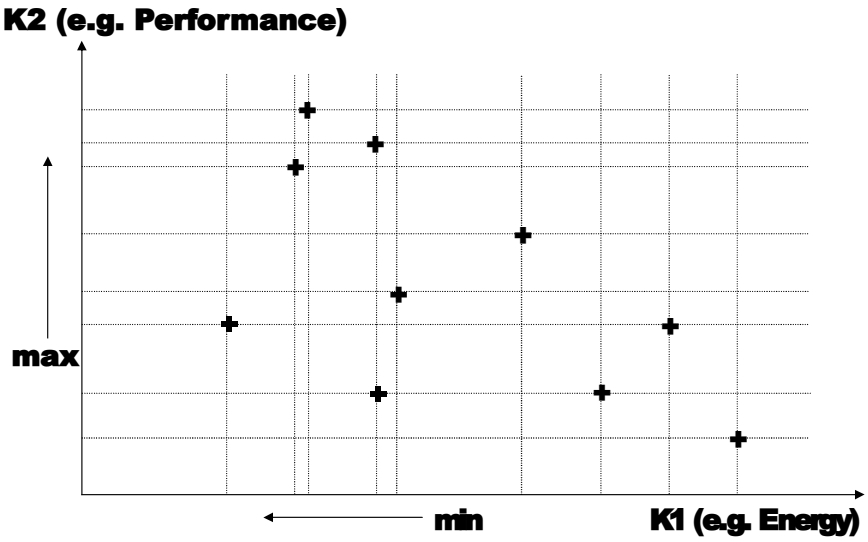
Hints: These assignments will be discussed at E23 OH14, from 10:15 am - 11:45 am on 24, Jan., 2017. You are not obligated to turn in the solutions.

1 Pareto Optimizations (2 Punkte)

- The following diagram reflects the evaluation of designs with respect to multiple criteria. We assume that we would like to maximize one criteria and minimize the other one. Please indicate the region which is dominated by design (1) (the region in which designs are “inferior” to design (1)). Also, indicate the region in which designs would dominate design (1) (the region in which designs are “superior” to design (1)).



- Suppose that design points marked “+” in the following diagram are given. Add the Pareto front to the diagram!



2 Hardware-Software Partitionierung (3 Punkte)

Nehmen Sie an, dass ein Task entweder komplett auf einem FPGA oder komplett auf dem Mikroprozessor ausgeführt werden kann. Nehmen Sie außerdem an, dass es keine Abhängigkeit zwischen den Tasks gibt. Wenn ein Task τ_i auf dem FPGA ausgeführt wird, benötigt er dafür B_i konfigurierbare logische Blöcke (configurable logical blocks (CLB)) und hat eine Ausführungszeit von F_i. Wenn der Task auf dem Mikroprozessor ausgeführt wird, ist seine Ausführungszeit C_i. Die Tasks die auf dem Mikroprozessor zugeordnet werden, können nur sequentiell ausgeführt werden, während die dem FPGA zugeordneten Tasks komplett parallel ausgeführt werden können.

Was ist die minimale Größe des FPGAs in Bezug auf die Anzahl der konfigurierbaren logischen Blöcke (CLBs) um alle Tasks $\{\tau_1, \tau_2, \ldots, \tau_n\}$ vor einer Deadline D zu beenden, wenn angenommen wird das alle Tasks zum Zeitpunkt 0 ankommen und damit ausführbar sind?

Formulieren Sie dieses Hardware-Software-Partitionierungs-Problem als ganzzahliges lineares Programm (integer linear programming (ILP)). Sie sollten ihr formuliertes ILP erklären. Es wird **NICHT** von Ihnen erwartet, dass Sie das ILP lösen.

3 Multiprocessor Partition Scheduling (3 Punkte)

Suppose that we are given M identical (homogeneous) processors. Each task τ_i is a periodic task with utilization U_i and period T_i. The utilization of a task τ_i is independent from which processor it is assigned to be executed. Please formulate an integer linear programming to *partition* the given task set $\{\tau_1, \tau_2, \ldots, \tau_n\}$ so that (1) a task τ_i can only and has to be assigned on one dedicated processor and (2) the tasks can be guaranteed to meet their deadlines under *rate monotonic* scheduling. The given tasks are assumed to be independent.

Hint: You can use the Liu and Layland bound.

4 Compositional WCET (2 Punkte)

Mr. Smart has derived the WCET 4 ms of a program on a (deterministic) platform at 1GHz CPU frequency. The system somehow is too powerful and can be slowed down to run at 500MHz CPU frequency (but all the rest of the platform remain the same) to reduce the power consumption. Mr. Smart concludes that the WCET under this new setting is 8 ms for this program. Is that a safe estimation? Is the estimation too pessimistic? If it is pessimistic, how can he improve the estimation? If it is not pessimistic, why not?

5 May Cache and Must Cache (2 Punkte)

Suppose that we have an LRU-cache with associativity 4.

- Is it possible to have less than 4 (different) entries in the Must Cache (Abstract Interpretation)?
- Is it possible to have more than 4 (different) entries in the Must Cache (Abstract Interpretation)?
- Is it possible to have less than 4 (different) entries in the May Cache (Abstract Interpretation)?
- Is it possible to have more than 4 (different) entries in the May Cache (Abstract Interpretation)?

Explain your answer.

6 Cache Analysis (2 Punkte)

Consider the following program:

```
read a;
read b;
read a;
if (a>b) {
  read c;
  read d;
} else {
  read e;
  read f;
}
read x;
read a;
```

(a) Perform May-Cache and Must-Cache analyses on this program, assuming an LRU (least-recently used) replacement policy with associativity 4 that is empty at the start of the program. Is it possible to determine whether the last access to “a” results in a cache hit or a cache miss?

(b) We now assume that the cache uses an FIFO (first-in-first-out) replacement policy instead. Could May-Cache and Must-Cache analyse determine whether the last access to “a” results in a cache hit or a cache miss if the cache is empty at the start of the program?

7 WCET for Systems with Scratchpad Memory (1 Punkt)

Explain the differences of WCET analysis when considering scratch pad memory (SPM) and Cache memory.

General Hints: